



Mamba RealTime Api Reference Manual

Automatically generated using doxygen

Contents

1	Data Structures	5
2	File List	5
3	buffer Struct Reference	5
4	MB_Image Struct Reference	6
4.1	Detailed Description	6
4.2	Field Documentation	6
4.2.1	depth	6
4.2.2	height	6
4.2.3	PIXARRAY	6
4.2.4	PLINES	6
4.2.5	width	6
5	MBRT_avcvidT Struct Reference	6
6	MBRT_Context Struct Reference	7
7	MBRT_v4l2vidT Struct Reference	7
8	MBRT_v4lvidT Struct Reference	7
9	MBRT_vidUnion Union Reference	8
10	mambaCommon.h File Reference	8
10.1	Detailed Description	9
10.2	Define Documentation	9
10.2.1	BIN_FILL_VALUE	9
10.2.2	GREY_FILL_VALUE	9
10.2.3	I32_FILL_VALUE	9
10.2.4	MB_LINE_COUNT	9
10.2.5	MB_LINE_OFFSET	9
10.2.6	MB_X_LEFT	9
10.2.7	MB_X_RIGHT	9
10.2.8	MB_Y_BOTTOM	9
10.2.9	MB_Y_TOP	10
10.3	Typedef Documentation	10
10.3.1	PIX32	10
10.3.2	PIX8	10
10.3.3	PLINE	10
10.3.4	PLINE32	10
10.3.5	Sint16	10
10.3.6	Sint32	10
10.3.7	Sint64	10
10.3.8	Sint8	10
10.3.9	Uint16	10
10.3.10	Uint32	10
10.3.11	Uint64	10
10.3.12	Uint8	10
10.4	Enumeration Type Documentation	10
10.4.1	MB_edgemode_t	10
10.4.2	MB_grid_t	10

11 mambaRTApi.h File Reference	11
11.1 Detailed Description	11
11.2 Enumeration Type Documentation	11
11.2.1 MBRT_eventcode	11
11.2.2 MBRT_vidType	11
11.3 Function Documentation	12
11.3.1 MBRT_CreateContext	12
11.3.2 MBRT_CreateDisplay	12
11.3.3 MBRT_CreateVideoAcq	12
11.3.4 MBRT_DestroyContext	12
11.3.5 MBRT_DestroyDisplay	12
11.3.6 MBRT_DestroyVideoAcq	12
11.3.7 MBRT_GetAcqFrameRate	12
11.3.8 MBRT_GetAcqSize	13
11.3.9 MBRT_GetImageFromAcq	13
11.3.10 MBRT_IconDisplay	13
11.3.11 MBRT_PaletteDisplay	13
11.3.12 MBRT_PollDisplay	13
11.3.13 MBRT_RecordEnd	13
11.3.14 MBRT_RecordImage	14
11.3.15 MBRT_RecordStart	14
11.3.16 MBRT_UpdateDisplay	14
12 mambaRTApi_loc.h File Reference	14
12.1 Detailed Description	15
12.2 Function Documentation	16
12.2.1 MBRT_CreateVideoAcq_avc	16
12.2.2 MBRT_CreateVideoAcq_v4l	16
12.2.3 MBRT_CreateVideoAcq_v4l2	16
12.2.4 MBRT_DestroyVideoAcq_avc	16
12.2.5 MBRT_DestroyVideoAcq_v4l	16
12.2.6 MBRT_DestroyVideoAcq_v4l2	16
12.2.7 MBRT_GetAcqFrameRate_avc	16
12.2.8 MBRT_GetAcqFrameRate_v4l	17
12.2.9 MBRT_GetAcqFrameRate_v4l2	17
12.2.10 MBRT_GetAcqSize_avc	17
12.2.11 MBRT_GetAcqSize_v4l	17
12.2.12 MBRT_GetAcqSize_v4l2	17
12.2.13 MBRT_GetImageFromAcq_avc	17
12.2.14 MBRT_GetImageFromAcq_v4l	18
12.2.15 MBRT_GetImageFromAcq_v4l2	18
12.3 Variable Documentation	18
12.3.1 context	18
13 MBRT_Context.c File Reference	18
13.1 Detailed Description	18
13.2 Function Documentation	18
13.2.1 MBRT_CreateContext	18
13.2.2 MBRT_DestroyContext	18
13.3 Variable Documentation	19
13.3.1 context	19
14 MBRT_Display.c File Reference	19
14.1 Detailed Description	19
14.2 Function Documentation	19
14.2.1 MBRT_CreateDisplay	19
14.2.2 MBRT_DestroyDisplay	19
14.2.3 MBRT_IconDisplay	20
14.2.4 MBRT_PaletteDisplay	20

14.2.5	MBRT_PollDisplay	20
14.2.6	MBRT_UpdateDisplay	20
15	MBRT_error.c File Reference	20
15.1	Detailed Description	21
15.2	Function Documentation	21
15.2.1	MBRT_StrErr	21
15.3	Variable Documentation	21
15.3.1	err_str	21
16	MBRT_error.h File Reference	21
16.1	Detailed Description	21
16.2	Enumeration Type Documentation	22
16.2.1	MBRT_errcode	22
16.3	Function Documentation	22
16.3.1	MBRT_StrErr	22
17	MBRT_Record.c File Reference	23
17.1	Detailed Description	23
17.2	Function Documentation	23
17.2.1	MBRT_RecordEnd	23
17.2.2	MBRT_RecordImage	23
17.2.3	MBRT_RecordStart	23
18	MBRT_VideoAcq.c File Reference	23
18.1	Detailed Description	24
18.2	Function Documentation	24
18.2.1	MBRT_CreateVideoAcq	24
18.2.2	MBRT_DestroyVideoAcq	24
18.2.3	MBRT_GetAcqFrameRate	24
18.2.4	MBRT_GetAcqSize	24
18.2.5	MBRT_GetImageFromAcq	24
19	MBRT_VideoAcq_avc.c File Reference	25
19.1	Detailed Description	25
19.2	Function Documentation	25
19.2.1	MBRT_CreateVideoAcq_avc	25
19.2.2	MBRT_DestroyVideoAcq_avc	25
19.2.3	MBRT_GetAcqFrameRate_avc	25
19.2.4	MBRT_GetAcqSize_avc	25
19.2.5	MBRT_GetImageFromAcq_avc	26
20	MBRT_VideoAcq_v4l.c File Reference	26
20.1	Detailed Description	26
20.2	Function Documentation	26
20.2.1	MBRT_CreateVideoAcq_v4l	26
20.2.2	MBRT_DestroyVideoAcq_v4l	26
20.2.3	MBRT_GetAcqFrameRate_v4l	26
20.2.4	MBRT_GetAcqSize_v4l	27
20.2.5	MBRT_GetImageFromAcq_v4l	27
21	MBRT_VideoAcq_v4l2.c File Reference	27
21.1	Detailed Description	27
21.2	Function Documentation	27
21.2.1	MBRT_CreateVideoAcq_v4l2	27
21.2.2	MBRT_DestroyVideoAcq_v4l2	28
21.2.3	MBRT_GetAcqFrameRate_v4l2	28
21.2.4	MBRT_GetAcqSize_v4l2	28
21.2.5	MBRT_GetImageFromAcq_v4l2	28

List of Figures

1	Image structure and variables	29
---	---	----

Data Structure Index

1 Data Structures

Here are the data structures with brief descriptions:

buffer	5
MB_Image	6
MBRT_avcvidT	6
MBRT_Context	7
MBRT_v4l2vidT	7
MBRT_v4lvidT	7
MBRT_vidUnion	8
File Index	

2 File List

Here is a list of all documented files with brief descriptions:

mambaCommon.h	8
mambaRTApi.h	11
mambaRTApi_loc.h	14
MBRT_Context.c	18
MBRT_Display.c	19
MBRT_error.c	20
MBRT_error.h	21
MBRT_Record.c	23
MBRT_VideoAcq.c	23
MBRT_VideoAcq_avc.c	25
MBRT_VideoAcq_v4l.c	26
MBRT_VideoAcq_v4l2.c	27
Data Structure Documentation	

3 buffer Struct Reference

Data Fields

- void * **start**
- size_t **length**

The documentation for this struct was generated from the following file:

- **mambaRTApi_loc.h**

4 MB_Image Struct Reference

```
#include <mambaCommon.h>
```

Data Fields

- **Uint32 width**
- **Uint32 height**
- **Uint32 depth**
- **PLINE * PLINES**
- **PIX8 * PIXARRAY**

4.1 Detailed Description

Images structure with the width, height and depth; the pixels array (PIXARRAY) and entry point array to each line of the image (PLINES)

4.2 Field Documentation

4.2.1 Uint32 MB_Image::depth

The depth of the image

4.2.2 Uint32 MB_Image::height

The height of the image

4.2.3 PIX8* MB_Image::PIXARRAY

pixel array

4.2.4 PLINE* MB_Image::PLINES

accessors to pixel lines

4.2.5 Uint32 MB_Image::width

The width of the image

The documentation for this struct was generated from the following file:

- **mambaCommon.h**

5 MBRT_avcvidT Struct Reference

Data Fields

- AVFormatContext * **format_ctx**
- AVCodecContext * **codec_ctx**
- int **videoStream**
- AVFrame * **frame**
- AVPacket **packet**

The documentation for this struct was generated from the following file:

- **mambaRTapi_loc.h**

6 MBRT__Context Struct Reference

Data Fields

- **MBRT__vidType** type
- int **fd**
- **MBRT__vidUnion** video
- **SDL_Surface** * **screen**
- **Uint32** **sz_x**
- **Uint32** **sz_y**
- **SDL_Color** **color_palette** [256]
- **SDL_Color** **standard_palette** [256]
- **Uint32** **isPalettized**
- **Uint32** **isFpsDisplayed**
- **Uint32** **old_call** [FPS_MEAN_SIZE]
- int **index_fps**
- **Uint32** **histo** [256]
- **Uint32** **isHistoDisplayed**
- **Uint8** **icon** [256]
- **Sint32** **isFullscreen**
- **Uint32** **isRecording**
- **AVFormatContext** * **rec_fmt_ctx**
- **AVFrame** * **pictureRGB**
- **AVFrame** * **picture**
- **struct SwsContext** * **img_convert_ctx**
- **uint8_t** * **video_outbuf**

The documentation for this struct was generated from the following file:

- **mambaRTapi_loc.h**

7 MBRT__v4l2vidT Struct Reference

Data Fields

- **struct buffer** * **buffers**
- unsigned int **n_buffers**
- int **w**
- int **h**

The documentation for this struct was generated from the following file:

- **mambaRTapi_loc.h**

8 MBRT__v4lvidT Struct Reference

Data Fields

- **PIX8** * **FRAMEBUFFER**
- **struct video_capability** **vcap**
- **struct video_window** **vwin**
- **struct video_picture** **vpic**
- **struct video_mmap** **vmmmap**
- **struct video_mbuf** **vmbuf**
- **struct video_channel** **vchan**

The documentation for this struct was generated from the following file:

- **mambaRTapi_loc.h**

9 MBRT_ vidUnion Union Reference

Data Fields

- `MBRT_v4lvidT v4l`
- `MBRT_v4l2vidT v4l2`
- `MBRT_avcvidT avc`

The documentation for this union was generated from the following file:

- `mambaRTapi_loc.h`

File Documentation

10 mambaCommon.h File Reference

```
#include <stdint.h>
```

Data Structures

- `struct MB_Image`

Defines

- `#define MB_X_LEFT(im) X_LEFT`
- `#define MB_X_RIGHT(im) X_RIGHT`
- `#define MB_Y_TOP(im) Y_TOP`
- `#define MB_Y_BOTTOM(im) Y_BOTTOM`
- `#define MB_LINE_COUNT(im) ((im->width*im->depth)/CHARBIT)`
- `#define MB_LINE_OFFSET(im) (MB_X_LEFT(im))`
- `#define BIN_FILL_VALUE(edge) ((edge==MB_FILLED_EDGE) ? UINT32_MAX:0)`
- `#define GREY_FILL_VALUE(edge) ((edge==MB_FILLED_EDGE) ? UINT32_MAX:0)`
- `#define I32_FILL_VALUE(edge) ((edge==MB_FILLED_EDGE) ? UINT32_MAX:0)`

Typedefs

- `typedef uint8_t Uint8`
- `typedef uint16_t Uint16`
- `typedef uint32_t Uint32`
- `typedef uint64_t Uint64`
- `typedef int8_t Sint8`
- `typedef int16_t Sint16`
- `typedef int32_t Sint32`
- `typedef int64_t Sint64`
- `typedef uint8_t PIX8`
- `typedef PIX8 * PLINE`
- `typedef uint32_t PIX32`
- `typedef PIX32 * PLINE32`

Enumerations

- `enum MB_grid_t { MB_HEXAGONAL_GRID = 1, MB_SQUARE_GRID = 0 }`
- `enum MB_edgemode_t { MB_EMPTY_EDGE = 0, MB_FILLED_EDGE = 1 }`

10.1 Detailed Description

Date:

31-03-2009

This file contains the various definitions, macro, struct that are commons between the various modules of the library

The copyright license of Mamba is reminded here :

Copyright (c) <2009>, <Nicolas BEUCHER and ARMINES for the Centre de Morphologie Mathématique(CMM), common research center to ARMINES and MINES Paristech>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

Except as contained in this notice, the names of the above copyright holders shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without their prior written authorization.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

10.2 Define Documentation

10.2.1 `#define BIN_FILL_VALUE(edge) ((edge==MB_FILLED_EDGE) ?
UINT32_MAX:0)`

How to fill the edge (binary bits images)

10.2.2 `#define GREY_FILL_VALUE(edge) ((edge==MB_FILLED_EDGE) ?
UINT32_MAX:0)`

How to fill the edge (8 bits images)

10.2.3 `#define I32_FILL_VALUE(edge) ((edge==MB_FILLED_EDGE) ?
UINT32_MAX:0)`

How to fill the edge (32 bits images)

10.2.4 `#define MB_LINE_COUNT(im) ((im->width*im->depth)/CHARBIT)`

Returns the size in bytes of an image line

10.2.5 `#define MB_LINE_OFFSET(im) (MB_X_LEFT(im))`

Returns the size in bytes of the line offset

10.2.6 `#define MB_X_LEFT(im) X_LEFT`

Getting image frame offset from left

10.2.7 `#define MB_X_RIGHT(im) X_RIGHT`

Getting image frame offset from right

10.2.8 `#define MB_Y_BOTTOM(im) Y_BOTTOM`

Getting image frame offset from bottom

10.2.9 `#define MB_Y_TOP(im) Y_TOP`

Getting image frame offset from top

10.3 Typedef Documentation

10.3.1 `typedef uint32_t PIX32`

Signed 32-bit pixels value type

10.3.2 `typedef uint8_t PIX8`

grey-scale pixels value type

10.3.3 `typedef PIX8* PLINE`

Pixels line pointers type

10.3.4 `typedef PIX32* PLINE32`

32-bit pixels line pointers type

10.3.5 `typedef int16_t Sint16`

Signed 16 bit value type

10.3.6 `typedef int32_t Sint32`

Signed 32 bit value type

10.3.7 `typedef int64_t Sint64`

Signed 64 bit value type

10.3.8 `typedef int8_t Sint8`

Signed 8 bit value type

10.3.9 `typedef uint16_t Uint16`

Unsigned 16 bit value type

10.3.10 `typedef uint32_t Uint32`

Unsigned 32 bit value type

10.3.11 `typedef uint64_t Uint64`

Unsigned 64 bit value type

10.3.12 `typedef uint8_t Uint8`

Unsigned 8 bit value type

10.4 Enumeration Type Documentation

10.4.1 `enum MB_edgemode_t`

enumerate for edge mode : either empty or filled

10.4.2 `enum MB_grid_t`

enumerate for grid : either square or hexagonal the value is supposed to give the number of directions allowed

11 mambaRTApi.h File Reference

```
#include "mambaCommon.h"
#include "MBRT_error.h"
```

Enumerations

- enum MBRT_ `eventcode` { `NO_EVENT`, `EVENT_CLOSE`, `EVENT_PROCESS`, `EVENT_PAUSE` }
- enum MBRT_ `vidType` { `V4L_TYPE`, `V4L2_TYPE`, `AVC_TYPE`, `NONE_TYPE` }

Functions

- MBRT_ `errcode` MBRT_ `CreateContext` (void)
- MBRT_ `errcode` MBRT_ `DestroyContext` (void)
- MBRT_ `errcode` MBRT_ `CreateVideoAcq` (char *device, MBRT_ `vidType` type)
- MBRT_ `errcode` MBRT_ `DestroyVideoAcq` (void)
- MBRT_ `errcode` MBRT_ `GetAcqSize` (int *acq_w, int *acq_h)
- MBRT_ `errcode` MBRT_ `GetAcqFrameRate` (double *ofps)
- MBRT_ `errcode` MBRT_ `GetImageFromAcq` (MB_ `Image` *dest)
- MBRT_ `errcode` MBRT_ `RecordStart` (char *filename)
- MBRT_ `errcode` MBRT_ `RecordEnd` (void)
- MBRT_ `errcode` MBRT_ `RecordImage` (MB_ `Image` *src)
- MBRT_ `errcode` MBRT_ `CreateDisplay` (int w, int h)
- MBRT_ `errcode` MBRT_ `DestroyDisplay` (void)
- MBRT_ `errcode` MBRT_ `UpdateDisplay` (MB_ `Image` *src, double wfps, double *ofps)
- MBRT_ `errcode` MBRT_ `PaletteDisplay` (UInt8 *palette)
- MBRT_ `errcode` MBRT_ `IconDisplay` (UInt8 *icon)
- MBRT_ `errcode` MBRT_ `PollDisplay` (MBRT_ `eventcode` *event_code)

11.1 Detailed Description

Date:

03-27-2009

This file contains the various definitions, global variables macro, struct and functions created for the library.

11.2 Enumeration Type Documentation

11.2.1 enum MBRT_ `eventcode`

Type definition for display event code

Enumerator:

- NO_EVENT* No events
- EVENT_CLOSE* Close event occurs when the user close the window or press esc inside it
- EVENT_PROCESS* Event to toggle on/off the process
- EVENT_PAUSE* Event to activate or deactivate the pause

11.2.2 enum MBRT_ `vidType`

Type definition for the acquisition device

Enumerator:

- V4L_TYPE* Video 4 Linux 1 API type device
- V4L2_TYPE* Video 4 Linux 2 API type device
- AVC_TYPE* Audio video codec API type device (video file)
- NONE_TYPE* No device

11.3 Function Documentation

11.3.1 `MBRT_errcode MBRT_CreateContext (void)`

Creates the context used throughout the library. The context describes the data displayed, the device used for acquisition, and contains general information needed by all the functions

11.3.2 `MBRT_errcode MBRT_CreateDisplay (int w, int h)`

Initializes SDL and creates the video display (SDL screen)

Parameters:

w width of the display (resolution)

h height of the display (resolution)

Returns:

An error code (NO_ERR if successful)

11.3.3 `MBRT_errcode MBRT_CreateVideoAcq (char * device, MBRT_vidType type)`

Fills the video acquisition structure with the parameters of the given device and initialize it.

Parameters:

device the video device (usually /dev/video0) or the video file.

type the type of device (e.g video4linux, video4linux2, avc ...)

Returns:

an error code (NO_ERR if successful)

11.3.4 `MBRT_errcode MBRT_DestroyContext (void)`

Destroys the context and all the data associated to it

11.3.5 `MBRT_errcode MBRT_DestroyDisplay (void)`

Destroys the video display (SDL screen) and quit SDL

Returns:

An error code (NO_ERR if successful)

11.3.6 `MBRT_errcode MBRT_DestroyVideoAcq (void)`

Closes the acquisition device and resets the structure

Returns:

NO_ERR if successful

11.3.7 `MBRT_errcode MBRT_GetAcqFrameRate (double * ofps)`

Returns the acquisition device default framerate.

Parameters:

fps the framerate in frame per second (output)

Returns:

NO_ERR if successful

11.3.8 MBRT_errcode MBRT_GetAcqSize (int * *acq_w*, int * *acq_h*)

Returns the acquisition device resolution.

Parameters:

acq_w the width (output)

acq_h the height (output)

Returns:

NO_ERR if successful

11.3.9 MBRT_errcode MBRT_GetImageFromAcq (MB_Image * *dest*)

Obtains an image form the acquisition device

Parameters:

dest the mamba image filled by the device

Returns:

NO_ERR if successful

11.3.10 MBRT_errcode MBRT_IconDisplay (Uint8 * *icon*)

Changes the small (16x16) icon (black and white) in the upper left corner of the display. This allows to inform the user of some events. The icon is an array of 256 integers. A 0 value will mean the pixel is not drawn, non 0 are drawn using the OSD color.

Parameters:

icon the 16x16 icon pixels array

Returns:

An error code (NO_ERR if successful)

11.3.11 MBRT_errcode MBRT_PaletteDisplay (Uint8 * *palette*)

Changes the palette associated with the display

Parameters:

palette an array containing the complete palette definition (256*3 integers)

Returns:

An error code (NO_ERR if successful)

11.3.12 MBRT_errcode MBRT_PollDisplay (MBRT_eventcode * *event_code*)

Handles event that have occurred in the display

Parameters:

event_code an integer representing a specific event (output)

Returns:

An error code (NO_ERR if successful)

11.3.13 MBRT_errcode MBRT_RecordEnd (void)

Ends the recording. Closes the codec and terminates the file.

Returns:

An error code (NO_ERR if successful)

11.3.14 `MBRT_errcode MBRT_RecordImage (MB_Image * src)`

Records an image

Parameters:

src the mamba image to record

Returns:

An error code (NO_ERR if successful)

11.3.15 `MBRT_errcode MBRT_RecordStart (char * filename)`

Starts the recording. This function activates the libavformat and libavcodec libraries to encode the image using MPEG2 codec (DVD format).

Parameters:

filename the path to the created video file

Returns:

An error code (NO_ERR if successful)

11.3.16 `MBRT_errcode MBRT_UpdateDisplay (MB_Image * src, double wfps, double * ofps)`

Updates the display with the content of a given Mamba image structure

Parameters:

src the image displayed

wfps input the desired framerate

ofps output the framerate

Returns:

An error code (NO_ERR if successful)

12 mambaRTApi_loc.h File Reference

```
#include "mambaRTApi.h"
#include <stdio.h>
#include <stdlib.h>
#include <stdint.h>
#include <unistd.h>
#include <string.h>
#include <libv4l2.h>
#include <libv4l1.h>
#include <linux/videodev.h>
#include <linux/videodev2.h>
#include <sys/mman.h>
#include <sys/file.h>
#include <sys/ioctl.h>
#include <SDL/SDL.h>
#include <libavcodec/avcodec.h>
#include <libavformat/avformat.h>
#include <libswscale/swscale.h>
```

Data Structures

- struct `MBRT__v4lvidT`
- struct `buffer`
- struct `MBRT__v4l2vidT`
- struct `MBRT__avcvidT`
- union `MBRT__vidUnion`
- struct `MBRT__Context`

Defines

- `#define MBRT__TITLE "Mamba RealTime"`
- `#define FRAME__COLOR 0xa00000`
- `#define FPS__VALUE__COLOR 0xa00000`
- `#define FPS__THICKNESS 3`
- `#define FPS__MEAN__SIZE 20`
- `#define HISTO__BLACKENING 60`
- `#define HISTO__COLOR 0xffff`
- `#define REC__SIZE 16`
- `#define REC__COLOR 0xb00000`

Functions

- `MBRT__errcode MBRT__CreateVideoAcq__v4l (char *device)`
- `MBRT__errcode MBRT__DestroyVideoAcq__v4l (void)`
- `MBRT__errcode MBRT__GetAcqSize__v4l (int *acq_w, int *acq_h)`
- `MBRT__errcode MBRT__GetAcqFrameRate__v4l (double *fps)`
- `MBRT__errcode MBRT__GetImageFromAcq__v4l (MB__Image *dest)`
- `MBRT__errcode MBRT__CreateVideoAcq__v4l2 (char *device)`
- `MBRT__errcode MBRT__DestroyVideoAcq__v4l2 (void)`
- `MBRT__errcode MBRT__GetAcqSize__v4l2 (int *acq_w, int *acq_h)`
- `MBRT__errcode MBRT__GetAcqFrameRate__v4l2 (double *fps)`
- `MBRT__errcode MBRT__GetImageFromAcq__v4l2 (MB__Image *dest)`
- `MBRT__errcode MBRT__CreateVideoAcq__avc (char *video_path)`
- `MBRT__errcode MBRT__DestroyVideoAcq__avc ()`
- `MBRT__errcode MBRT__GetAcqSize__avc (int *acq_w, int *acq_h)`
- `MBRT__errcode MBRT__GetAcqFrameRate__avc (double *fps)`
- `MBRT__errcode MBRT__GetImageFromAcq__avc (MB__Image *dest)`

Variables

- `MBRT__Context * context`

12.1 Detailed Description

Date:

03-27-2009

This file contains the various definitions, global variables macro, struct and functions that are shared between components of the library but are not meant to be exported to the outside world.

12.2 Function Documentation

12.2.1 MBRT_errcode MBRT_CreateVideoAcq_avc (char * *video_path*)

Opens a video file as a video acquisition device using the advanced video codec (AVC) library.

Parameters:

video_path path to the video file (supported codec depend on your local implementation of the libavcodec)

Returns:

an error code (NO_ERR if successful)

12.2.2 MBRT_errcode MBRT_CreateVideoAcq_v4l (char * *device*)

Fills the video acquisition (V4L) structure with the parameters of the given device and initializes it.

Parameters:

device the video device (usually /dev/video0)

Returns:

an error code (NO_ERR if successful)

12.2.3 MBRT_errcode MBRT_CreateVideoAcq_v4l2 (char * *device*)

Fills the video acquisition (V4L2) structure with the parameters of the given device and initializes it.

Parameters:

device the video device (usually /dev/video0)

Returns:

an error code (NO_ERR if successful)

12.2.4 MBRT_errcode MBRT_DestroyVideoAcq_avc ()

Closes the video file playback (AVC) and resets the structure

Returns:

NO_ERR if successful

12.2.5 MBRT_errcode MBRT_DestroyVideoAcq_v4l (void)

Closes the acquisition device (V4L) and resets the structure

Returns:

NO_ERR if successful

12.2.6 MBRT_errcode MBRT_DestroyVideoAcq_v4l2 (void)

Closes the acquisition device (V4L2) and resets the structure

Returns:

NO_ERR if successful

12.2.7 MBRT_errcode MBRT_GetAcqFrameRate_avc (double * *fps*)

Returns the acquisition device default framerate (AVC).

Parameters:

fps the framerate in frame per second (output)

Returns:

NO_ERR if successful

12.2.8 MBRT_errcode MBRT_GetAcqFrameRate_v4l (double * *ofps*)

Returns the acquisition device default framerate (V4L).

Parameters:

ofps the framerate in frame per second (output)

Returns:

NO_ERR if successful

12.2.9 MBRT_errcode MBRT_GetAcqFrameRate_v4l2 (double * *ofps*)

Returns the acquisition device default framerate (V4L2).

Parameters:

ofps the framerate in frame per second (output)

Returns:

NO_ERR if successful

12.2.10 MBRT_errcode MBRT_GetAcqSize_avc (int * *acq_w*, int * *acq_h*)

Returns the video played size (AVC).

Parameters:

acq_w the width (output)

acq_h the height (output)

Returns:

NO_ERR if successful

12.2.11 MBRT_errcode MBRT_GetAcqSize_v4l (int * *acq_w*, int * *acq_h*)

Returns the acquisition device resolution (V4L).

Parameters:

acq_w the width (output)

acq_h the height (output)

Returns:

NO_ERR if successful

12.2.12 MBRT_errcode MBRT_GetAcqSize_v4l2 (int * *acq_w*, int * *acq_h*)

Returns the acquisition device resolution (V4L2).

Parameters:

acq_w the width (output)

acq_h the height (output)

Returns:

NO_ERR if successful

12.2.13 MBRT_errcode MBRT_GetImageFromAcq_avc (MB_Image * *dest*)

Obtains an image from the video (AVC)

Parameters:

dest the mamba image filled by the device

Returns:

NO_ERR if successful

12.2.14 MBRT_errcode MBRT_GetImageFromAcq_v4l (MB_Image * *dest*)

Obtains an image from the acquisition device (V4L)

Parameters:

dest the mamba image filled by the device

Returns:

NO_ERR if successful

12.2.15 MBRT_errcode MBRT_GetImageFromAcq_v4l2 (MB_Image * *dest*)

Obtains an image from the acquisition device (V4L2)

Parameters:

dest the mamba image filled by the device

Returns:

NO_ERR if successful

12.3 Variable Documentation

12.3.1 MBRT_Context* context

Structure holding the complete context information (display, device, ...)

13 MBRT_Context.c File Reference

```
#include "mambaRTapi_loc.h"
```

Functions

- MBRT_errcode MBRT_CreateContext ()
- MBRT_errcode MBRT_DestroyContext ()

Variables

- MBRT_Context * context = NULL

13.1 Detailed Description

Author:

Nicolas Beucher

Date:

17-04-2010

13.2 Function Documentation

13.2.1 MBRT_errcode MBRT_CreateContext (void)

Creates the context used throughout the library. The context describes the data displayed, the device used for acquisition, and contains general information needed by all the functions

13.2.2 MBRT_errcode MBRT_DestroyContext (void)

Destroys the context and all the data associated to it

13.3 Variable Documentation

13.3.1 MBRT_Context* context = NULL

Structure holding the complete context information (display, device, ...)

14 MBRT_Display.c File Reference

```
#include "mambaRTapi_loc.h"
```

Defines

- #define **RED**(pix) (*(pix+2))
- #define **GREEN**(pix) (*(pix+1))
- #define **BLUE**(pix) (*pix)

Functions

- MBRT_errcode MBRT_CreateDisplay (int w, int h)
- MBRT_errcode MBRT_DestroyDisplay ()
- MBRT_errcode MBRT_UpdateDisplay (MB_Image *src, double wfps, double *ofps)
- MBRT_errcode MBRT_IconDisplay (Uint8 *icon)
- MBRT_errcode MBRT_PaletteDisplay (Uint8 *palette)
- MBRT_errcode MBRT_PollDisplay (MBRT_eventcode *event_code)

14.1 Detailed Description

Author:

Nicolas Beucher

Date:

03-29-2009

14.2 Function Documentation

14.2.1 MBRT_errcode MBRT_CreateDisplay (int w, int h)

Initializes SDL and creates the video display (SDL screen)

Parameters:

w width of the display (resolution)

h height of the display (resolution)

Returns:

An error code (NO_ERR if successful)

14.2.2 MBRT_errcode MBRT_DestroyDisplay (void)

Destroys the video display (SDL screen) and quit SDL

Returns:

An error code (NO_ERR if successful)

14.2.3 MBRT_errcode MBRT_IconDisplay (Uint8 * *icon*)

Changes the small (16x16) icon (black and white) in the upper left corner of the display. This allows to inform the user of some events. The icon is an array of 256 integers. A 0 value will mean the pixel is not drawn, non 0 are drawn using the OSD color.

Parameters:

icon the 16x16 icon pixels array

Returns:

An error code (NO_ERR if successful)

14.2.4 MBRT_errcode MBRT_PaletteDisplay (Uint8 * *palette*)

Changes the palette associated with the display

Parameters:

palette an array containing the complete palette definition (256*3 integers)

Returns:

An error code (NO_ERR if successful)

14.2.5 MBRT_errcode MBRT_PollDisplay (MBRT_eventcode * *event_code*)

Handles event that have occurred in the display

Parameters:

event_code an integer representing a specific event (output)

Returns:

An error code (NO_ERR if successful)

14.2.6 MBRT_errcode MBRT_UpdateDisplay (MB_Image * *src*, double *wfps*, double * *ofps*)

Updates the display with the content of a given Mamba image structure

Parameters:

src the image displayed

wfps input the desired framerate

ofps output the framerate

Returns:

An error code (NO_ERR if successful)

15 MBRT_error.c File Reference

```
#include "MBRT_error.h"
```

Functions

- char * MBRT_StrErr (MBRT_errcode error_nb)

Variables

- char * err_str []

15.1 Detailed Description

Author:

Nicolas Beucher

Date:

3-28-2009

15.2 Function Documentation

15.2.1 `char* MBRT_StrErr (MBRT_errcode error_nb)`

Returns an explanation of the error code

15.3 Variable Documentation

15.3.1 `char* err_str[]`

Error value interpretation

16 MBRT_error.h File Reference

Enumerations

- enum MBRT_errcode {
 - NO_ERR, ERR_CANT_CREATE_CONTEXT, ERR_INVLD_CTX, ERR_INIT_DISPLAY,
 - ERR_FORMAT_DISPLAY, ERR_LOCK_DISPLAY, ERR_INVALID_DISPLAY, ERR_VID,
 - ERR_OPEN_VID, ERR_V4L_VID, ERR_V4L2_VID, ERR_STRM_VID,
 - ERR_CAP_VID, ERR_FMT_VID, ERR_PAL_VID, ERR_VID_TYPE,
 - ERR_DEPTH, ERR_SIZE, ERR_AVC_VID_OPEN, ERR_AVC_STREAM_INFO,
 - ERR_AVC_NO_VID_STREAM, ERR_AVC_NO_CODEEC, ERR_AVC_CODEEC_OPEN,
 - ERR_AVC_FRAME_ALLOC,
 - ERR_AVC_DECODING, ERR_AVC_REC_FORMAT, ERR_AVC_REC_FMTCTX,
 - ERR_AVC_REC_STREAM,
 - ERR_AVC_REC_PARAM_SET, ERR_AVC_REC_NO_CODEEC, ERR_AVC_REC_CODEEC_OPEN,
 - ERR_AVC_REC_PICT_ALLOC,
 - ERR_AVC_REC_FILE_OPEN, ERR_AVC_REC_ENCODE, ERR_AVC_REC_INV_CTX, ERR_UNKNOWN }

Functions

- `char * MBRT_StrErr (MBRT_errcode error_nb)`

16.1 Detailed Description

Date:

3-28-2009

This file contains the complete liste of error code returned by the API functions.

16.2 Enumeration Type Documentation

16.2.1 enum MBRT_errcode

Type definition for error code

Enumerator:

NO_ERR Value returned by function when no error was encountered.

ERR_CANT_CREATE_CONTEXT Cannot create (memory allocation) context

ERR_INVLD_CTX Context is not properly initialized

ERR_INIT_DISPLAY Init SDL display error

ERR_FORMAT_DISPLAY SDL display cannot support requested format

ERR_LOCK_DISPLAY Locking screen for updating failure

ERR_INVALID_DISPLAY Incorrect display (not initialized) error

ERR_VID Video acquisition module error

ERR_OPEN_VID Cannot open video device

ERR_V4L_VID Not a video for linux device

ERR_V4L2_VID Not a video for linux 2 device

ERR_STRM_VID Device does not support streaming

ERR_CAP_VID Cannot obtain device resolution capabilities

ERR_FMT_VID Cannot set device format

ERR_PAL_VID Unsupported palette format

ERR_VID_TYPE Type of the video acquisition is incorrect

ERR_DEPTH The depth of the mamba image given in argument is incompatible

ERR_SIZE The size of the mamba image given in argument is incompatible

ERR_AVC_VID_OPEN Cannot open the video file

ERR_AVC_STREAM_INFO Cannot retrieve video stream information

ERR_AVC_NO_VID_STREAM No video stream present in video file

ERR_AVC_NO_CODEC Cannot find appropriate video decoder

ERR_AVC_CODEC_OPEN Cannot open the video codec

ERR_AVC_FRAME_ALLOC Cannot allocate the video frame

ERR_AVC_DECODING Cannot decode video frame

ERR_AVC_REC_FORMAT Cannot find format for recording

ERR_AVC_REC_FMTCTX Cannot allocate format context for recording

ERR_AVC_REC_STREAM Cannot create stream for recording

ERR_AVC_REC_PARAM_SET Invalid output format parameters for recording

ERR_AVC_REC_NO_CODEC Cannot find appropriate video decoder for recording

ERR_AVC_REC_CODEC_OPEN Cannot open the video codec for recording

ERR_AVC_REC_PICT_ALLOC Cannot allocate picture for recording

ERR_AVC_REC_FILE_OPEN Cannot open the output file for recording

ERR_AVC_REC_ENCODE Cannot encode the image for recording

ERR_AVC_REC_INV_CTX Invalid recording context (not started)

ERR_UNKNOWN Unknown error

16.3 Function Documentation

16.3.1 char* MBRT_StrErr (MBRT_errcode error_nb)

Returns an explanation of the error code

17 MBRT_Record.c File Reference

```
#include "mambaRTapi_loc.h"
```

Functions

- `MBRT_errcode MBRT_RecordStart (char *filename)`
- `MBRT_errcode MBRT_RecordEnd ()`
- `MBRT_errcode MBRT_RecordImage (MB_Image *src)`

17.1 Detailed Description

Author:

Nicolas Beucher

Date:

17-04-2010

17.2 Function Documentation

17.2.1 `MBRT_errcode MBRT_RecordEnd (void)`

Ends the recording. Closes the codec and terminates the file.

Returns:

An error code (NO_ERR if successful)

17.2.2 `MBRT_errcode MBRT_RecordImage (MB_Image * src)`

Records an image

Parameters:

src the mamba image to record

Returns:

An error code (NO_ERR if successful)

17.2.3 `MBRT_errcode MBRT_RecordStart (char * filename)`

Starts the recording. This function activates the libavformat and libavcodec libraries to encode the image using MPEG2 codec (DVD format).

Parameters:

filename the path to the created video file

Returns:

An error code (NO_ERR if successful)

18 MBRT_VideoAcq.c File Reference

```
#include "mambaRTapi_loc.h"
```

Functions

- `MBRT_errcode MBRT_CreateVideoAcq (char *device, MBRT_vidType type)`
- `MBRT_errcode MBRT_DestroyVideoAcq ()`
- `MBRT_errcode MBRT_GetAcqSize (int *acq_w, int *acq_h)`
- `MBRT_errcode MBRT_GetAcqFrameRate (double *ofps)`
- `MBRT_errcode MBRT_GetImageFromAcq (MB_Image *dest)`

18.1 Detailed Description

Author:

Nicolas Beucher

Date:

03-27-2009

18.2 Function Documentation

18.2.1 `MBRT_errcode MBRT_CreateVideoAcq (char * device, MBRT_vidType type)`

Fills the video acquisition structure with the parameters of the given device and initialize it.

Parameters:

device the video device (usually /dev/video0) or the video file.

type the type of device (e.g video4linux, video4linux2, avc ...)

Returns:

an error code (NO_ERR if successful)

18.2.2 `MBRT_errcode MBRT_DestroyVideoAcq (void)`

Closes the acquisition device and resets the structure

Returns:

NO_ERR if successful

18.2.3 `MBRT_errcode MBRT_GetAcqFrameRate (double * fps)`

Returns the acquisition device default framerate.

Parameters:

fps the framerate in frame per second (output)

Returns:

NO_ERR if successful

18.2.4 `MBRT_errcode MBRT_GetAcqSize (int * acq_w, int * acq_h)`

Returns the acquisition device resolution.

Parameters:

acq_w the width (output)

acq_h the height (output)

Returns:

NO_ERR if successful

18.2.5 `MBRT_errcode MBRT_GetImageFromAcq (MB_Image * dest)`

Obtains an image form the acquisition device

Parameters:

dest the mamba image filled by the device

Returns:

NO_ERR if successful

19 MBRT_VideoAcq_avc.c File Reference

```
#include "mambaRTapi_loc.h"
```

Functions

- `MBRT_errcode MBRT_CreateVideoAcq_avc (char *video_path)`
- `MBRT_errcode MBRT_DestroyVideoAcq_avc ()`
- `MBRT_errcode MBRT_GetAcqSize_avc (int *acq_w, int *acq_h)`
- `MBRT_errcode MBRT_GetAcqFrameRate_avc (double *ofps)`
- `MBRT_errcode MBRT_GetImageFromAcq_avc (MB_Image *dest)`

19.1 Detailed Description

Author:

Nicolas Beucher

Date:

04-01-2010

19.2 Function Documentation

19.2.1 `MBRT_errcode MBRT_CreateVideoAcq_avc (char * video_path)`

Opens a video file as a video acquisition device using the advanced video codec (AVC) library.

Parameters:

video_path path to the video file (supported codec depend on your local implementation of the libavcodec)

Returns:

an error code (NO_ERR if successful)

19.2.2 `MBRT_errcode MBRT_DestroyVideoAcq_avc ()`

Closes the video file playback (AVC) and resets the structure

Returns:

NO_ERR if successful

19.2.3 `MBRT_errcode MBRT_GetAcqFrameRate_avc (double * ofps)`

Returns the acquisition device default framerate (AVC).

Parameters:

fps the framerate in frame per second (output)

Returns:

NO_ERR if successful

19.2.4 `MBRT_errcode MBRT_GetAcqSize_avc (int * acq_w, int * acq_h)`

Returns the video played size (AVC).

Parameters:

acq_w the width (output)

acq_h the height (output)

Returns:

NO_ERR if successful

19.2.5 MBRT_errcode MBRT_GetImageFromAcq_avc (MB_Image * *dest*)

Obtains an image from the video (AVC)

Parameters:

dest the mamba image filled by the device

Returns:

NO_ERR if successful

20 MBRT_VideoAcq_v4l.c File Reference

```
#include "mambaRTapi_loc.h"
```

Functions

- MBRT_errcode MBRT_CreateVideoAcq_v4l (char *device)
- MBRT_errcode MBRT_DestroyVideoAcq_v4l ()
- MBRT_errcode MBRT_GetAcqSize_v4l (int *acq_w, int *acq_h)
- MBRT_errcode MBRT_GetAcqFrameRate_v4l (double *ofps)
- MBRT_errcode MBRT_GetImageFromAcq_v4l (MB_Image *dest)

20.1 Detailed Description

Author:

Nicolas Beucher

Date:

04-07-2009

20.2 Function Documentation

20.2.1 MBRT_errcode MBRT_CreateVideoAcq_v4l (char * *device*)

Fills the video acquisition (V4L) structure with the parameters of the given device and initializes it.

Parameters:

device the video device (usually /dev/video0)

Returns:

an error code (NO_ERR if successful)

20.2.2 MBRT_errcode MBRT_DestroyVideoAcq_v4l (void)

Closes the acquisition device (V4L) and resets the structure

Returns:

NO_ERR if successful

20.2.3 MBRT_errcode MBRT_GetAcqFrameRate_v4l (double * *ofps*)

Returns the acquisition device default framerate (V4L).

Parameters:

fps the framerate in frame per second (output)

Returns:

NO_ERR if successful

20.2.4 MBRT_errcode MBRT_GetAcqSize_v4l (int * acq_w, int * acq_h)

Returns the acquisition device resolution (V4L).

Parameters:

- acq_w* the width (output)
- acq_h* the height (output)

Returns:

NO_ERR if successful

20.2.5 MBRT_errcode MBRT_GetImageFromAcq_v4l (MB_Image * dest)

Obtains an image from the acquisition device (V4L)

Parameters:

- dest* the mamba image filled by the device

Returns:

NO_ERR if successful

21 MBRT_VideoAcq_v4l2.c File Reference

```
#include "mambaRTapi_loc.h"
```

Defines

- #define CLEAR(x) memset (&(x), 0, sizeof (x))

Functions

- MBRT_errcode MBRT_CreateVideoAcq_v4l2 (char *device)
- MBRT_errcode MBRT_DestroyVideoAcq_v4l2 ()
- MBRT_errcode MBRT_GetAcqSize_v4l2 (int *acq_w, int *acq_h)
- MBRT_errcode MBRT_GetAcqFrameRate_v4l2 (double *ofps)
- MBRT_errcode MBRT_GetImageFromAcq_v4l2 (MB_Image *dest)

21.1 Detailed Description

Author:

Nicolas Beucher

Date:

04-07-2009

21.2 Function Documentation

21.2.1 MBRT_errcode MBRT_CreateVideoAcq_v4l2 (char * device)

Fills the video acquisition (V4L2) structure with the parameters of the given device and initializes it.

Parameters:

- device* the video device (usually /dev/video0)

Returns:

an error code (NO_ERR if successful)

21.2.2 MBRT_errcode MBRT_DestroyVideoAcq_v4l2 (void)

Closes the acquisition device (V4L2) and resets the structure

Returns:

NO_ERR if successful

21.2.3 MBRT_errcode MBRT_GetAcqFrameRate_v4l2 (double * *ofps*)

Returns the acquisition device default framerate (V4L2).

Parameters:

fps the framerate in frame per second (output)

Returns:

NO_ERR if successful

21.2.4 MBRT_errcode MBRT_GetAcqSize_v4l2 (int * *acq_w*, int * *acq_h*)

Returns the acquisition device resolution (V4L2).

Parameters:

acq_w the width (output)

acq_h the height (output)

Returns:

NO_ERR if successful

21.2.5 MBRT_errcode MBRT_GetImageFromAcq_v4l2 (MB_Image * *dest*)

Obtains an image from the acquisition device (V4L2)

Parameters:

dest the mamba image filled by the device

Returns:

NO_ERR if successful

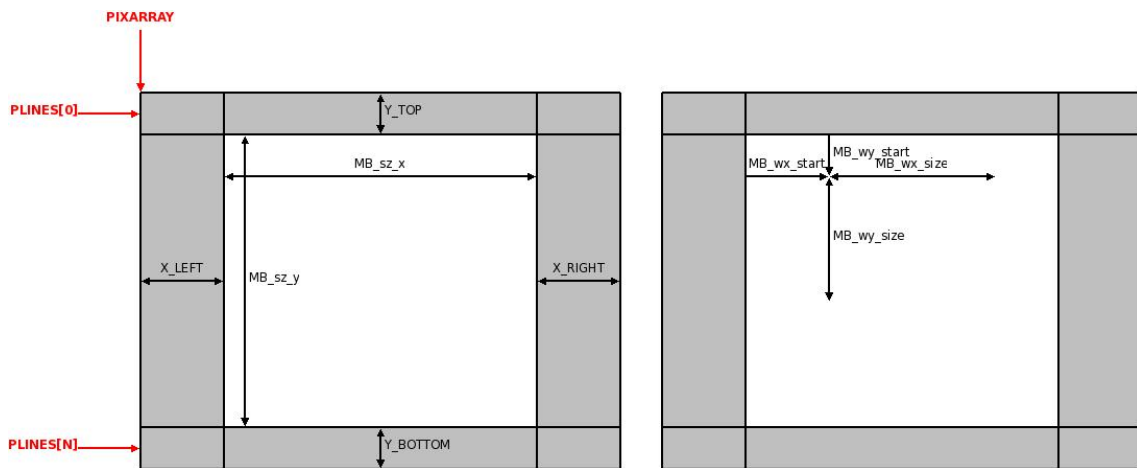


Figure 1: Image structure and variables